Reeb Graphs Through Local Binary Patterns

Ines Janusch and Walter G. Kropatsch

Pattern Recognition and Image Processing Group
Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria
`{ines,krw}@prip.tuwien.ac.at`

Abstract. This paper presents an approach to derive critical points of a shape, the basis of a Reeb graph, using a combination of a medial axis skeleton and features along this skeleton. A Reeb graph captures the topology of a shape. The nodes in the graph represent critical points (positions of change in the topology), while edges represent topological persistence. We present an approach to compute such critical points using Local Binary Patterns. For one pixel the Local Binary Pattern feature vector is derived comparing this pixel to its neighbouring pixels in an environment of a certain radius. We start with an initial segmentation and a medial axis representation. Along this axis critical points are computed using Local Binary Patterns with the radius, defining the neighbouring pixels, set a bit larger than the radius according to the medial axis transformation. Critical points obtained in this way form the node set in a Reeb graph, edges are given through the connectivity of the skeleton. This approach aims at improving the representation of flawed segmented data. In the same way segmentation artefacts, as for example single pixels representing noise, may be corrected based on this analysis.

Keywords: Reeb Graphs; Local Binary Patterns; Local Features; Critical Points; Shape Representation; Image Segmentation

1 Introduction

Reeb graphs capture a shape's topology: the connected components (the connectivity of the shape) are represented by the edges, while positions of change in topology are represented by nodes in the graph [1]. Reeb graphs are for example used as a tool of skeletonisation in [2], a tool of segmentation in [3] or a tool of shape analysis in [4]. Moreover, the compact shape description provided by a Reeb graph may be used for shape comparison and retrieval.

For a Reeb graph representation critical points are usually computed on the shape according to a Morse function [1]. The obtained Reeb graph is dependent on the applied Morse function and its properties. In contrast, we present a novel approach to derive the edges of a Reeb graph through the topology (connectivity) captured by a medial axis, the nodes are computed based on local features at certain positions in the shape.

In [4] we presented an analysis of roots for the purpose of plant phenotyping using Reeb graphs. Our results showed that Reeb graphs are suitable for such an analysis. However, the sensitivity of this representation to segmentation errors is likely to falsify results. To reduce the influence of segmentation errors, the computation of the Reeb graph should

not be based solely on segmented data.

In order to analyse a shape using a Morse function, the input image needs to be binary segmented. Such a pre-processing may introduce artefacts e.g. spurious branches in the graph representation. Post-processing methods applied to the graph representation, for example graph pruning, may be used to discard spurious branches.

Instead of a post-processing procedure we propose an approach that is based on an initial shape representation according to a segmentation of the input data. For this initial representation the medial axis is used. It is obtained as the centers of maximally inscribed circles on a shape. The medial axis (in 2D and 3D) as well as the more sophisticated curve skeletons [5] are used as compact shape descriptor as they capture the topological characteristics of a shape. However, the medial axis is sensitive to noise due to segmentation artefacts. Thus, the skeleton is only used as a first representation to guide the computation of a Reeb graph representation.

Based on the skeleton, critical points that form the node set of a Reeb graph, are computed using Local Binary Patterns (LBPs) [6] centered along the skeleton. For the computation of critical points according to a Morse function, Morse conditions need to be kept. One such Morse condition requires the Morse function values of two different critical points to differ in order to derive a unique graph representation. However, when representing shapes in a discrete space (e.g. 2D pixel or 3D voxel space) this condition is likely to be violated as discussed in [7].

LBPs analyse and represent an image region, the neighbourhood at certain radius, around a pixel. A common application for LBPs is given by texture classification as originally presented in [8]. Moreover, LBPs have been used for face recognition in [9]. We use LBPs to computed nodes of a Reeb graph. When determined as the critical points of a Morse function, these nodes may be classified according to the different changes in topology they represent. Here, the different types of nodes correspond with the neighbourhood configurations that can be represented by an LBP[1]. LBPs allow to base the representation on the original unsegmented data. Starting from an initial segmentation and an initial skeleton representation LBPs may be computed on the unsegmented image. Segmentation artefacts may be detected and corrected in this way.

The rest of the paper is structured as follows: Section 2 provides an introduction to LBPs, Section 3 to Reeb graphs. The computation of a Reeb graph through LBPs is defined in Section 4. Results obtained are discussed in Section 5. Section 6 concludes the paper and gives an outlook to future work.

2   Principle of Local Binary Patterns

LBPs were introduced by Ojala, Pietikäinen and Harwood in 1994 [8] as a tool of texture classification. Due to its computational simplicity and its robustness to spurious color gradients e.g. due to lighting conditions, LBPs are popular texture operators.

A simple example how LBPs work is shown in Figure 1: The center pixel is compared

---

[1] Although identical names are used for Reeb graph node types and LBP types an identical meaning is not guaranteed (e.g. an LBP of type saddle may not represent a saddle node in a Reeb graph). We indicate Reeb graph node types by [R], LBP types by [L] (e.g. saddle[R], saddle[L])

$x_1 \geq c$	$x_2 < c$	$x_3 < c$		1	0	0		2^0	2^1	2^2		**LBP** = 10001111
$x_8 \geq c$	c	$x_4 < c$		1	c	0		2^7	c	2^3		$= 2^0 + 2^4 + 2^5 +$
$x_7 \geq c$	$x_6 \geq c$	$x_5 \geq c$		1	1	1		2^6	2^5	2^4		$2^6 + 2^7$ $= \mathbf{241}$

(a) center pixel and neighbourhood   (b) comparison result   (c) neighbourhood pattern   (d) LBP operator for center pixel c

Fig. 1: Simple LBP computation.

(a) max^L (b) min^L (c) plateau^L (d) slope^L (e) saddle^L

Fig. 2: Neighbourhood configuration detected by LBPs. The red circle indicates the neighbourhood used in the LBP computation for the pixel at its center.

to its subsampled neighbourhood. The relations of this comparison are stored as a bit pattern: In case the value of a neighbouring pixel is larger than or equal to the value of the center pixel its bit is set to 1 otherwise to 0. The neighbourhood pattern is encoded as the position of each neighbourhood pixel in a binary data item [8].

Moreover, the configuration of the neighbourhood around a pixel encodes the local topology. The region may be a (local) maximum^L (the bit pattern contains only 0s), a (local) minimum^L (the bit pattern contains only 1s), a plateau^L (the bit pattern contains only 1s, but all pixels of the region have the same gray value), a slope^L (the bit pattern of the region contains one connected component of 1s and one connected component of 0s) or a saddle^L point otherwise [10]. Figure 2 shows examples for these region configurations that may be encoded by LBPs. The approach presented in this paper uses the different region configurations to derive critical points of a shape in order to represent it using a Reeb graph.

LBPs are not only defined for the eight immediate neighbours of a pixel. These eight direct neighbours of a center pixel are at radius 1 from the center pixel but the radius at which an LBP operator is computed may also be larger than 1.

3   Morse Theory and Reeb Graphs

Reeb graphs describe the topological structure of a shape (e.g. 2D or 3D content) as the connectivity of its level sets [11]. A shape is analysed according to a Morse function to derive a Reeb graph. Two common Morse functions are the height function and the geodesic distance. The nodes of a Reeb graph correspond to critical points computed on a shape according to a Morse function. At critical points the topology of the analysed shape changes, thus the number of connected components in the level-set changes.

Edges connecting critical points represent the connected components and thus describe topological persistence.

A point $p = (a, b)$ of a function $f(x, y)$ is called a *critical point* if both derivatives $f_x(a, b)$ and $f_y(a, b)$ are equal to 0 or if one of these partial derivatives does not exist. Such a critical point p is called degenerate if the determinant of the Hessian matrix at that point is zero, otherwise it is called non-degenerate (or Morse) critical point [12].

A Morse functions is defined in the continuous domain as follows:

A smooth, real-valued function $f : M \rightarrow \mathbb{R}$ is called a Morse function if it satisfies the following conditions for a manifold M with or without boundary:

– *M1*: all critical points of f are non-degenerate and lie inside M,
– *M2*: all critical points of f restricted to the boundary of M are non-degenerate,
– *M3*: for all pairs of distinct critical points p and q, $f(p) \neq f(q)$ must hold [13].

Although originally defined in the continuous domain, Reeb graphs have been extended to the discrete domain:

– Two point sets are connected if there exists a pair of points (one point of each point set) with a distance between these two points below a fixed threshold.
– If all non-empty subsets of a point set, as well as its complements, are connected, such a point set is called *connective*.
– A group of points that have the same Morse function value and that form a connective point set, is called a *level-set curve* [3].

The nodes in a discrete Reeb graph represent level-set curves, the edges connect two adjacent level-set curves, therefore the underlying point sets are connected [3].

In 2D three types of nodes in a Reeb graph correspond to critical points: minima^R, maxima^R or saddles^R [13]. Minimum^R and maximum^R nodes are of degree 1, due to the conditions for Morse functions (especially condition $M3$) saddle^R nodes are of degree 3. An example Reeb graph containing all possible types of nodes is shown in Figure 3.
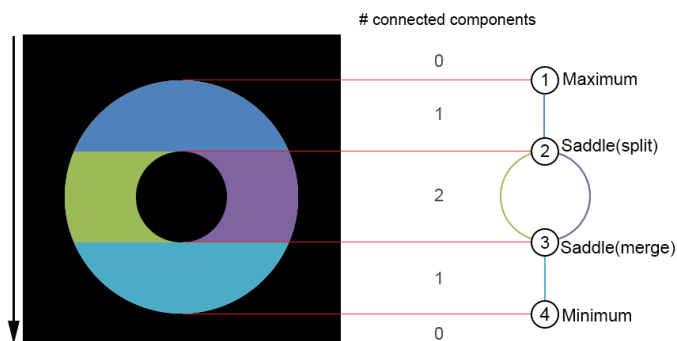

Fig. 3: Reeb graph of a ring according to the height function; the foreground region is colour labeled according to the connected components.

4   Computation of Reeb Graphs based on LBPs

Reeb graphs are derived on binary segmented 2D or 3D data using an analysis based on a Morse function. Different Morse functions applied to the same image may result in different Reeb graphs. Moreover, the computed Reeb graph depends on the properties of the Morse function used. A height function for example is not rotational invariant. Due to the rotational invariance property of the medial axis, Reeb graphs computed according to the approach presented in this paper are rotational invariant (apart from artefacts due to the discrete pixel space).

In order to analyse unsegmented data local descriptors, as for example LBPs, may identify the critical points directly on this data. For Reeb graphs based on a Morse function Morse conditions [13] apply. When computing the critical points (the nodes in a Reeb graph) according to LBPs, the following conditions apply: a critical point is determined by an LBP of type maximum^L, minimum^L, slope^L or saddle^L as described in Section 2. For LBPs of type saddle^L the neighbourhood configuration of the LBP may not be divided into more than six connected components (therefore three foreground regions, as saddle^R nodes are of degree 3). LBPs of type plateau^L do not represent critical points but regular points which do not represent any change in topology.

The approach presented here works on a segmented image as an input. The computation may be extended to unsegmented data. Nevertheless, a segmented image may still be needed as a first input to guide the computation of the critical points. In case the position of the critical points can be estimated (e.g. in video data based on the position in a previous frame) the segmentation is not necessary.

The computation of a Reeb graph according to LBPs in general works as follows:

1. initial binary segmentation and medial axis representation of foreground
2. computation of LBPs along skeleton pixels
3. determination of critical points
4. connection of critical points according to skeleton to obtain Reeb graph

Theorem 1. *This approach determines a Reeb graph representing the foreground shape.*

Proof. A conventional Reeb graph is defined as a topological graph which describes the evolution of level-sets of a real valued function on a manifold. The medial axis captures the topology of a shape. It is used in the presented approach to guide the graph computation and to define the connectivity. The obtained graph therefore is a topological graph. Our proposed method follows the pixels along a skeleton. The geodesic distance (along the skeleton, starting from an arbitrary skeleton pixel) serves as the function analysing the shape. Although, here one level-set only consists of single skeleton pixels at a certain distance to the starting point, the evolution of the level-sets is just as well described by the branching points and end points of the skeleton.

Changes of topology (e.g. an increase in the number of connected components due to a branch) are detected at the boundary of the shape when computing a Reeb graph according to a common Morse function (e.g. the height function or the geodesic distance). The medial axis guides our approach; critical points are positioned on the skeleton, therefore inside the shape. Our identification of critical points further considers LBPs along this skeleton. These LBPs are computed according to the medial axis radii, plus an increase of ϵ. Therefore, the boundary of the shape is taken into account.   □
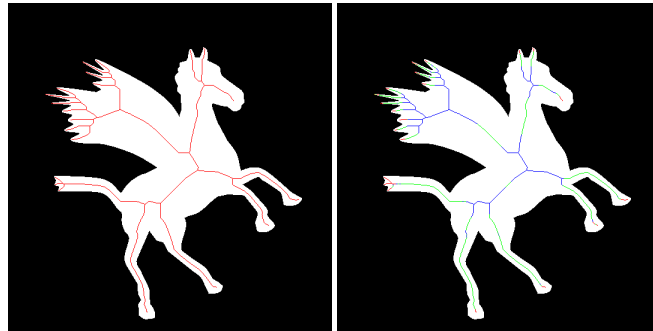
4.1 Initial Skeleton

An initial binary segmentation of the input image is needed to compute the medial axis for the foreground region. This axis is formed by the centers of maximal circles that cover the shape completely. Therefore, the medial axis implicitly provides a measure of width, for each point p_i along the medial axis the radius r_i of the inscribed maximal circle (the distance to the boundary) is known [14]. Figure 4a shows a segmented example image (from the the mythological creatures database [15]) together with the medial axis representing it.

4.2 LBPs along Skeleton

LBPs are computed for each skeleton pixel. According to the radius r_i stored with each skeleton pixel p_i, the LBP is computed for each p_i with a radius of $r_i * 1.5$. This enlargement by $\epsilon = 50\%$ was experimentally determined. ϵ may be adjusted according to the desired output, as it regulates the detection of spurious branches. It serves as Nyquist limit, as branches smaller than ϵ are discarded by this approach.

This radius is likely to be 15 pixel or more for the images in our dataset and in general. Therefore, it may not be possible to store the final LBP operator: For a radius of 15 pixel the LBP is (in our case) computed based on 64 neighbours along a circle of radius 15. As described in Section 2 the final LBP operator is obtained by converting the binary data item representing the neighbourhood of a pixel to the decimal system. This may result in numbers larger than 2^{64}, which cannot be represented in most programming languages. However, this LBP operator is not needed for the presented approach. Instead of computing LBP operators, we only compute the type of LBP neighbourhood configuration for each skeleton pixel. Figure 4b shows the skeleton pixels colour labeled according to the LBP neighbourhood configuration around these pixels.

In contrast to the LBP neighbourhood configurations shown in Figure 2 we only encounter the following three neighbourhood configurations (shown in Figure 5) when

(a) Binary segmented input image and its medial axis drawn in red.

(b) LBPs: red = slope^L, green = ridge^L, blue = branch^L.

Fig. 4: Computation of LBPs along the skeleton.

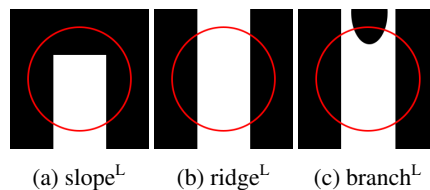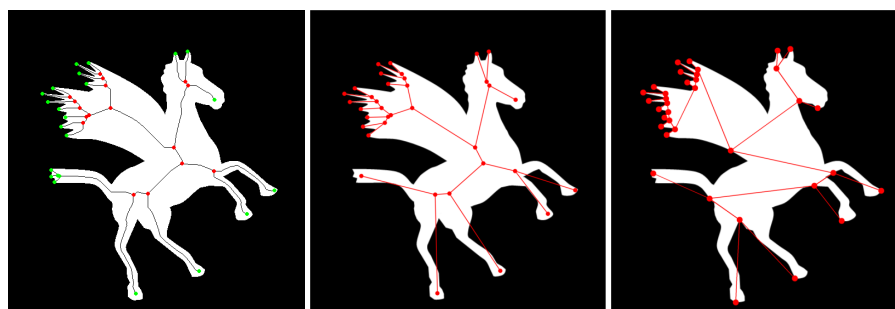(a) slope^L (b) ridge^L (c) branch^L

Fig. 5: Possible LBP neighbourhood configuration along skeleton pixels of a binary segmented image (LBP radius according to the medial axis radius but enlarged).

computing LBPs along a skeleton of a binary segmented image: slope^L, ridge^L (special case of saddle^L) and branch^L (again special case of saddle^L). They are determined as follows: the LBP bit pattern according to the pixels along the LBP radius consists of one connected component of 0s and one connected component of 1s for a slope^L, two connected components each for a ridge^L and three each for a branch^L.

4.3   Critical Points according to LBPs

To determine the position of critical points, the skeleton pixels are analysed according to the type of LBP and the number of neighbouring skeleton pixels. A critical point of type saddle^R is detected as skeleton pixel for which the LBP shows a branch^L and which has three neighbouring skeleton pixels. Critical points of type minimum^R/maximum^R are given as skeleton pixels with a corresponding LBP of type slope^L and only one neighbouring skeleton pixel. Skeleton pixels with an LBP indicating a ridge^L and two neighbouring skeleton pixels correspond to regular points. At such positions nodes of degree two may be added along an edge in a Reeb graph. However, as nodes of degree two do not describe any changes in topology, they are typically disregarded in a topological representation.

(a) Nodes based on LBPs: red = saddle^R, green = min^R/max^R. (b) Final Reeb graph based on LBPs along the skeleton. (c) Reeb graph using geodesic distance as Morse function.

Fig. 6: Computation of the Reeb graph according to the LBPs along the skeleton and according to the geodesic distance as Morse function.

In case the LBP and the number of neighbouring skeleton pixels do not correspond according to the above given definition, artefacts may be detected: In case a skeleton pixel with three neighbouring skeleton pixels has an LBP of a type different from branch^L, small curvature changes along the boundary may have introduced a spurious branch in the skeleton. By computing the LBP using the enlarged medial axis radius, small spurious branches can be detected and discarded. Artefacts are further caused by circular curvature changes along the boundary that may assign an LBP of type branch^L to a skeleton pixel with only two neighbouring skeleton pixels. Such pixels are corrected to be not represented as a critical point but as a regular node in the Reeb graph (note that regular nodes are discarded in the final graph representation).

Figure 6a shows the critical points compute for the skeleton and the LBPs in Figure 4. Depending on the size of the LBP radius at a certain skeleton pixel, branches may be detected as spurious branches. In the example image this is visible at the tail of the mythological horse. Here a saddle^R node (red) was labeled as minimum^R/maximum^R node (green). Such branches are discarded in the final graph representation (Figure 6b).

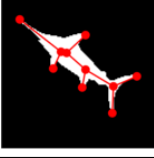### 4.4   Reeb Graph according to Skeleton and LBPs

The critical points are used as nodes in the Reeb graph. Edges are determined by the skeleton: The skeleton is traced from each branch point (saddle^R) to all neighbouring nodes. The resulting graph for the example image is shown in Figure 6b. For comparison Figure 6c shows a Reeb graph obtained using an alternative approach based on the geodesic distance as Morse function [4]. Apart from a difference in the positions of the nodes, the numbers of the represented connected components are equivalent.

5   Results

The approach presented in this paper shows work in progress. The results are therefore preliminary results.

Our approach was tested on the 99 images of the shape database presented in [16] and on the 15 images of the mythological creatures database [15]. Out of these 114 images the obtained Reeb graph correctly represents the topology for 60 images. In case the shape was not correctly represented, this can be ascribed to the graph pruning that is automatically included and dependent on the chosen LBP radius. Moreover, compact shapes with a circular boundary line may corrupt the representation. Table 1 shows Reeb graphs computed using the presented approach for 10 images of the evaluated database [16]. For simplicity the edges of the Reeb graph are drawn as straight lines although their precise geometry can be taken from the medial axis. For image 6 the head of the animal is not represented in the final graph . Due to the circular contour line the according skeleton segment was incorrectly detected as a spurious branch and therefore rejected. For image 10 no graph representation could be found, as well in this case the circular contour line rejected a branch in the skeleton as a true branch. Such representational artefacts may be avoided by computing more than one LBP operator according to different radii around one skeleton pixel, as well as computing the LBP operators on an unsegmented grayscale image instead of the segmented binary image.

Table 1: Reeb graph representation obtained for silhouette images.

However, this Reeb graph representation, just as any other topological graph representation is best used for elongated, branched or articulated objects. Compact shapes with a smooth, rounded contour line (circular, no elongation and no dents along it) hold only little characteristic topological information (no significant branches) with therefore little discriminative power. Thus, also the representational power of a topological graph representing such a shape is limited. Compact shapes do not present suitable applications for Reeb graphs in general, but are well suited for shock graphs [16].

6 Conclusion and Future Work

The presented approach does not analyse the binary segmented input image using a Morse function to compute a Reeb graph. Instead the computation of the Reeb graph is based on LBPs along a skeleton. In this way critical points, which form the node set of the Reeb graph, and the connectivity of the nodes (the edges) representing the shape's topology can be derived. The results show that this approach derives Reeb graph representations for shapes that may be well represented using a graph (for example articulated objects).

In contrast to Morse functions, which need to be evaluated for every pixel of the shape, this new approach determines the Reeb graph by evaluating a much smaller number of locations: only the axis points. A future goal is to further limit these evaluation positions (e.g. to use the end points and branching points of the axis only).

The silhouette images that were used in the experiments only present a first test dataset. For future work the intended input data are grayscale images. An initial segmentation may still be needed in order to derive the skeleton which guides the Reeb graph computation. Nevertheless, the LBP computation may be performed on the unsegmented data. In this way segmentation artefacts may be detected in the graph and discarded. The segmentation may even be corrected based on the observed LBP operators.

References

1. Biasotti, S., Giorgi, D., Spagnuolo, M., Falcidieno, B.: Reeb graphs for shape analysis and applications. Theoretical Computer Science **392**(1-3) (February 2008) 5–22
2. Ge, X., Safa, I.I., Belkin, M., Wang, Y.: Data skeletonization via Reeb graphs. In Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 24. (2011) 837–845
3. Werghi, N., Xiao, Y., Siebert, J.: A functional-based segmentation of human body scans in arbitrary postures. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **36**(1) (2006) 153–165
4. Janusch, I., Kropatsch, W.G., Busch, W., Ristova, D.: Representing roots on the basis of reeb graphs in plant phenotyping. In: ECCV 2014 Workshop on Computer Vision Problems in Plant Phenotyping. (2014) in press
5. Nyström, I., Di Baja, G.S., Svensson, S.: Curve skeletonization by junction detection in surface skeletons. In: Visual Form 2001. Volume 2059 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 229–238
6. Pietikäinen, M.: Image analysis with local binary patterns. In: Image Analysis. Volume 3540 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2005) 115–118
7. Janusch, I., Kropatsch, W.G., Busch, W.: Reeb graph based examination of root development. In: Proceedings of the 19th Computer Vision Winter Workshop. (Feb 2014) 43–50
8. Ojala, T., Pietikäinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on. Volume 1. (Oct 1994) 582–585
9. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Computer Vision - ECCV 2004. Volume 3021 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 469–481
10. Gonzalez-Diaz, R., Kropatsch, W., Cerman, M., Lamar, J.: Characterizing configurations of critical points through lbp. In: SYNASC 2014 Workshop on Computational Topology in Image Context. (2014)
11. EL Khoury, R., Vandeborre, J.P., Daoudi, M.: 3D mesh Reeb graph computation using commute-time and diffusion distances. In: Proceedings SPIE: Three-Dimensional Image Processing (3DIP) and Applications II. Volume 8290. (2012) 82900H–82900H–10
12. Bott, R.: Lectures on Morse theory, old and new. Bulletin of the American Mathematical Society **7**(2) (1982) 331–358
13. Doraiswamy, H., Natarajan, V.: Efficient algorithms for computing Reeb graphs. Computational Geometry **42**(67) (August 2009) 606–616
14. Lee, D.T.: Medial axis transformation of a planar shape. Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI-4**(4) (July 1982) 363–369
15. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Monographs in Computer Science. Springer New York (2009)
16. Sebastian, T., Klein, P., Kimia, B.: Recognition of shapes by editing shock graphs. In: Computer Vision, IEEE International Conference on. Volume 1., IEEE Computer Society (2001) 755–755